

INTEGRACIÓN DE UN PLUGIN A UN NAVEGADOR WEB PARA ANÁLISIS DE VULNERABILIDADES EN JAVASCRIPT

Rafael Enrique Rodriguez Rodriguez
Daniel Camilo Escobar Ravelo

Director
Luis Felipe Herrera Quintero Ph.D

Codirector
Jaime A. Chavarriaga Ph.D (c)

Resumen

En la actualidad la seguridad informática es una prioridad tanto de las grandes, medianas y pequeñas empresas, por este motivo se han desarrollado técnicas para mejorar la seguridad dentro de los lenguajes de programación[1] y así evitar la propagación de vulnerabilidades dentro de los códigos de las aplicaciones y páginas web. Por ello en este trabajo se plantea la integración de un Plug-in a un navegador web para análisis de vulnerabilidades en JavaScript.

Planteamiento del Problema

El problema está centrado analizar y verificar las vulnerabilidades de código JavaScript [2], e indagar específicamente en lo relacionado con los ataques informáticos y la manera de contrarrestarlos.

Objetivos

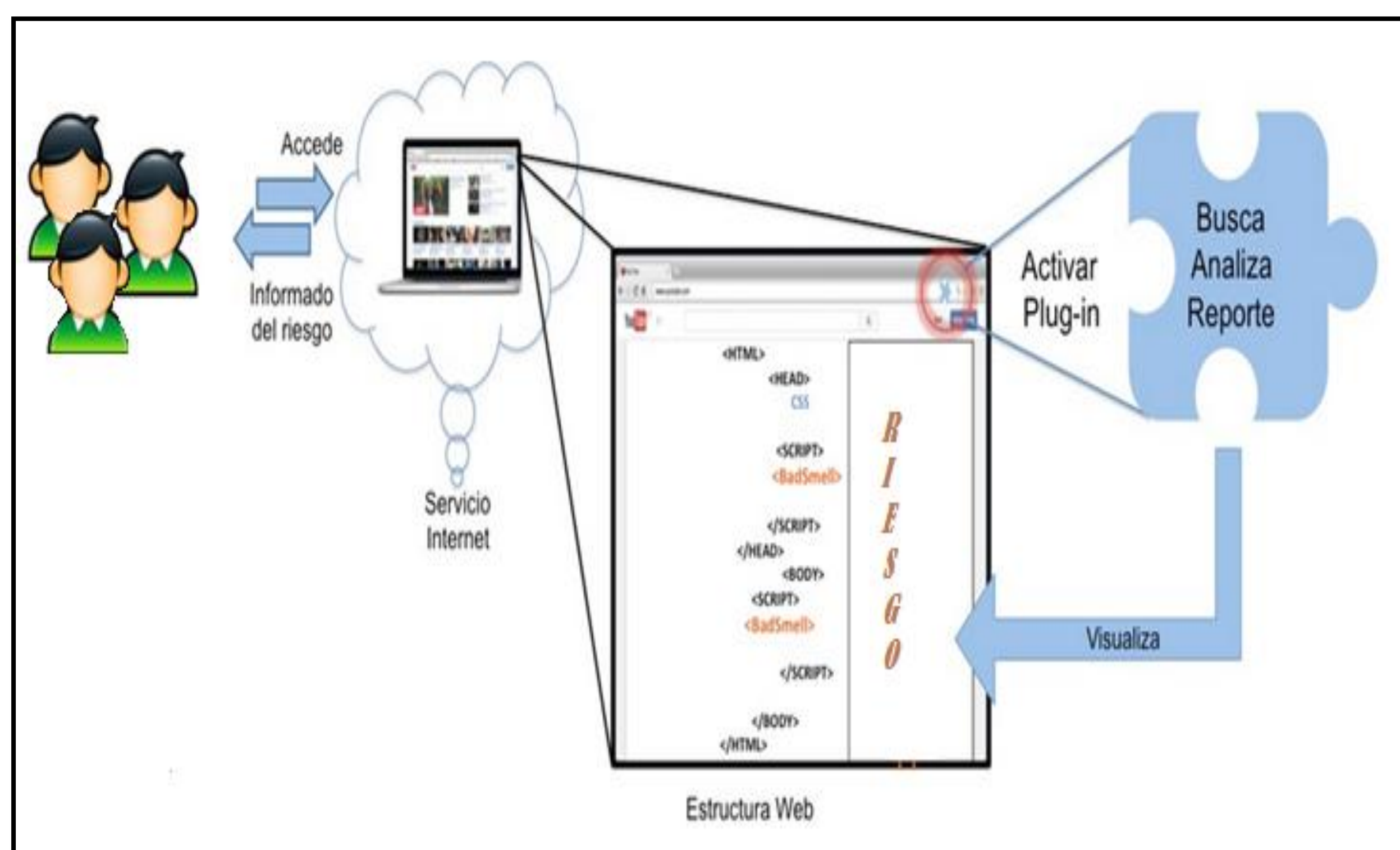
Objetivo general

- Integrar un Plug-in a un navegador web para analizar código vulnerable en JavaScript.

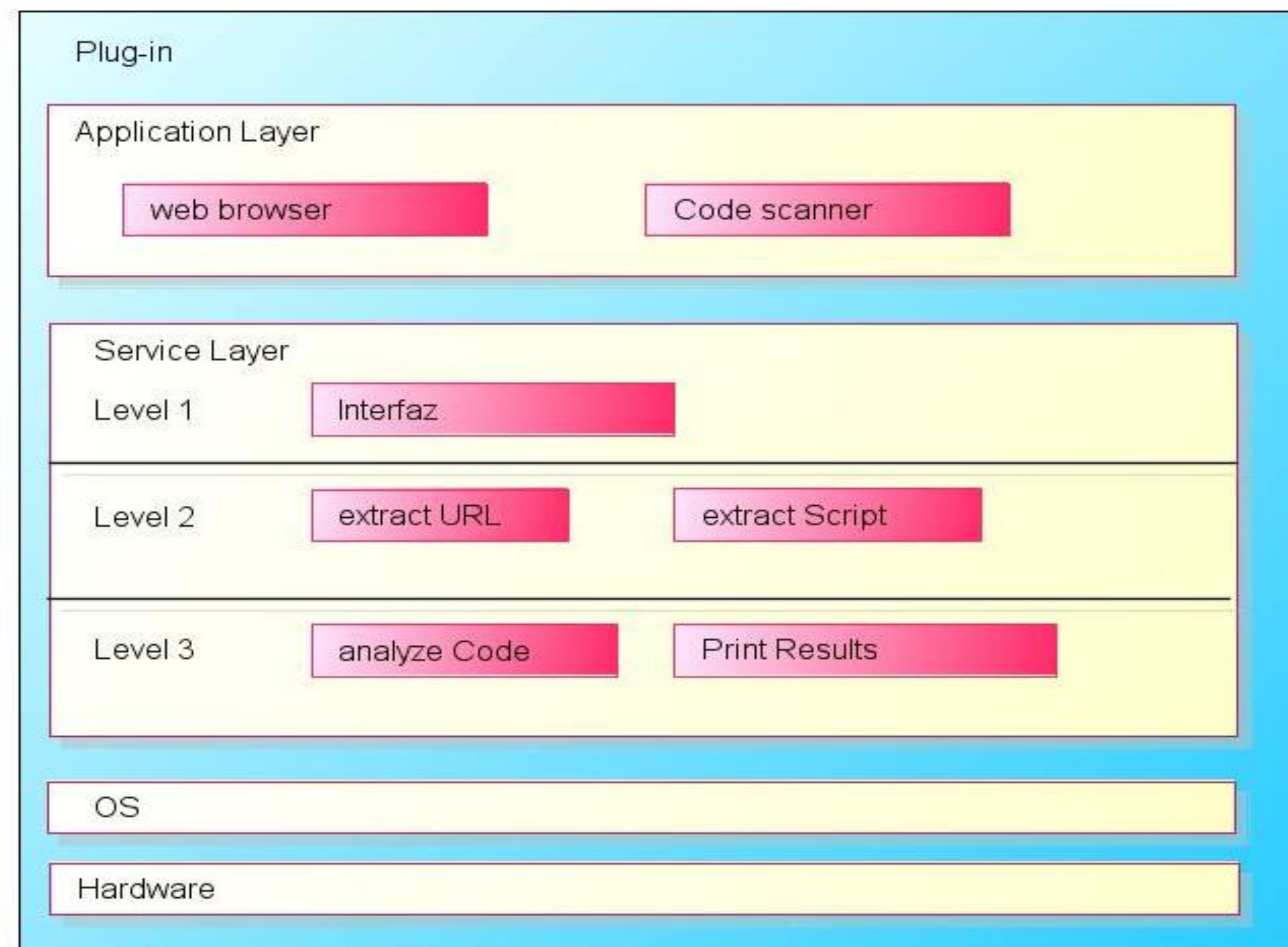
Objetivos específicos

- Revisar el estado del arte sobre la seguridad de los navegadores web y los peligros para el cliente al ejecutar JavaScript.
- Diseñar un Plug-in como mecanismo para detectar código vulnerable.
- Implementar y poner a prueba el Plug-in diseñado para la detección de código vulnerable.
- Realizar análisis de los resultados del Plug-in.

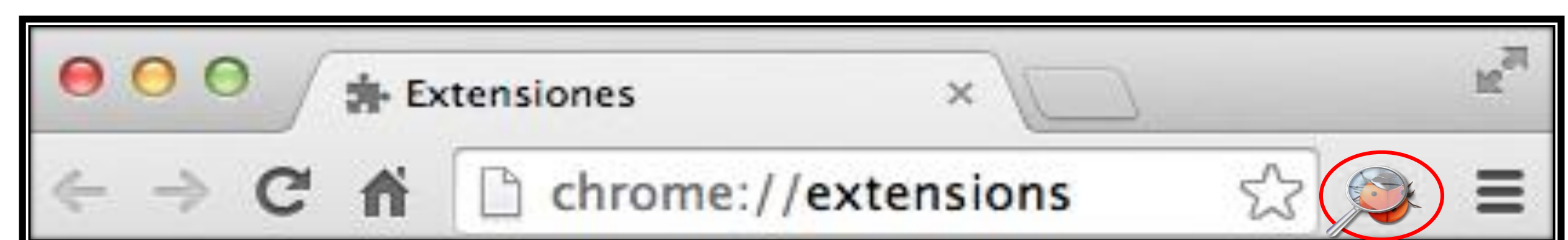
Propuesta del Plug-in



Arquitectura del Plug-in

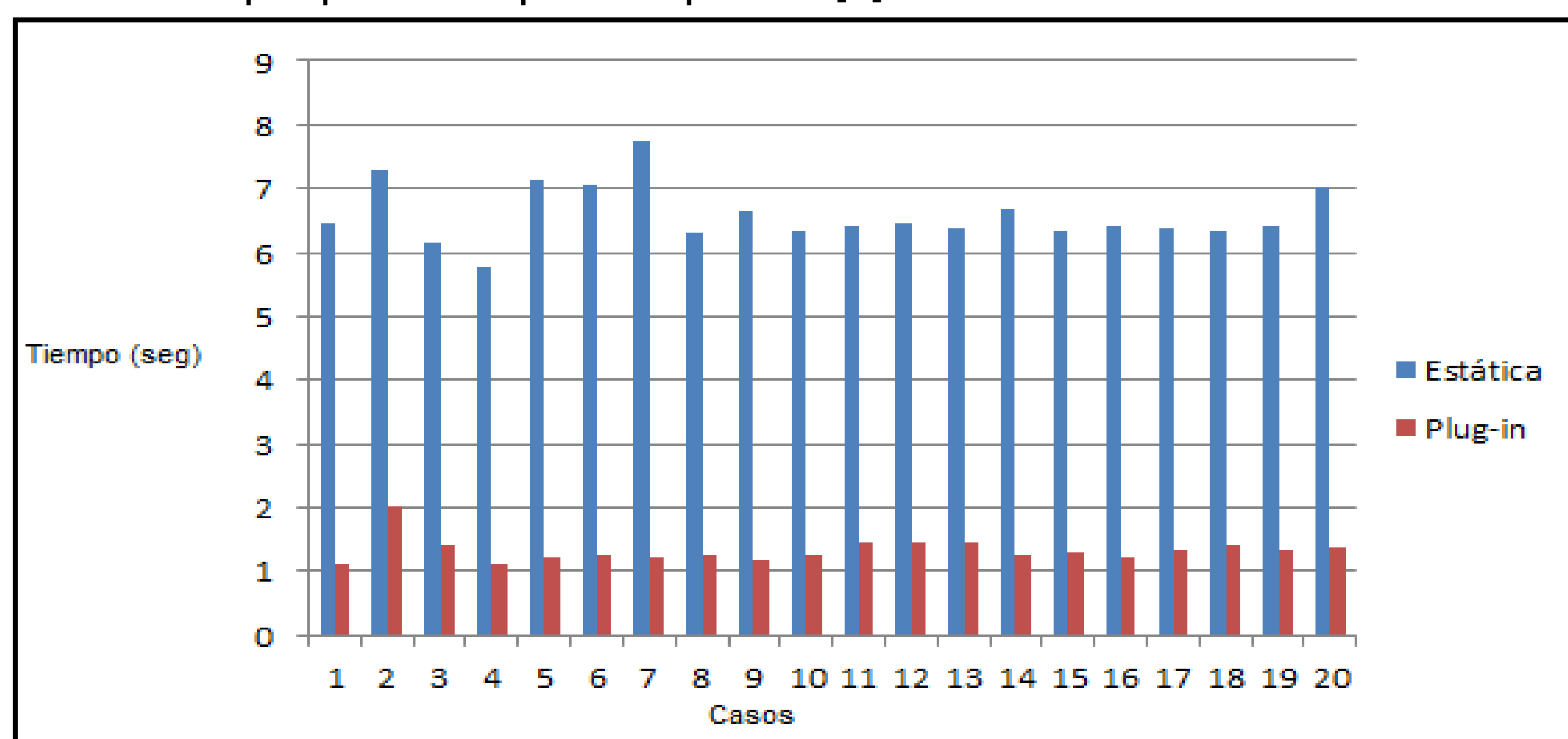


Implementación



Análisis de Resultados

Se presentan pruebas de rendimiento, usando la herramienta estática de JsPrime [3] y la implementación del Plug-in, con el fin de validar el tiempo de análisis y lo que tardaría un usuario en el análisis de este código sin la disponibilidad del Plug-in. Usando lo 20 casos propuestos para Jsprime [4].



Conclusiones

- Se toma ventaja del escáner de código JsPrime
- Se logra un acoplamiento entre JsPrime y el Plug-in para analizar vulnerabilidades en JavaScript
- Se reduce tiempo de análisis implementado el Plug-in propuesto
- Se identificaron los principales BadSmells, que afectan la seguridad de la páginas web

Referencias:

- [1]. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Source Code Security Analyzers*: NIST [online]. Aviable: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html
- [2]. BIELOVA, Natalia. "Survey on Security Policies JavaScript and enforcement mechanisms in a web browser". NRIA Rennes Bretagne Atlantique Campus Universitario de Beaulieu. 2013. 1-33.
- [3]. DAS PATNAIK, Nishant. SABYASACHI SAHOO, Sarathi. "JavaScript static analysis of security made easy with JSPrime". ECMAScript. 2012. 1-5.
- [4]DAS PATNAIK, Nishant .GoogleChrome. *JavaScript APIs*: Google Chrome [online]. Aviable: https://developer.chrome.com/extensions/api_index.